Time Complexity of the Ford-Fulkerson Algorithm

- For the case of real capacities, the Ford-Fulkerson algorithm can run forever
  - Moreover, it can converge to a suboptimal value
- For the case of integer capacities, we have seen that the number of iterations is upper bounded by the value of a maximum flow
  - Does this upper bound imply that Ford-Fulkerson runs in polynomial time?

Time Complexity of Ford-Fulkerson (Integer Capacities)

- Consider a flow network $G = (V, E)$ with $V = \{s, a, b, t\}$ and where the edges in E with positive capacity are (s, a), (s, b), (a, b), (a, t), and (b, t)
  - All edges have capacity K except for edge (a, b), which has capacity 1
  - The value of a maximum flow is $2K$
  - If we augment along the length-2 paths s, a, t and s, b, t, the algorithm terminates with a maximum flow after two iterations
  - However, if we always augment along a length-3 augmenting path, the algorithm runs for $2K$ iterations
  - The size of the instance is $\Theta(\log_2 K)$ bits
  - Thus a $\Theta(K)$ running time is exponential in the input size

The Edmonds-Karp Maximum Flow Algorithm

- A variant of the Ford-Fulkerson algorithm in which we always augment along a shortest augmenting path
  - The running time of each iteration is unchanged, since we can use BFS to find a shortest augmenting path in linear time
- We will prove that the Edmonds-Karp algorithm runs in polynomial time, even for real capacities
- Recall that the algorithm maintains a flow f at each iteration
- For any vertex v in V, let $d(v, i)$ denote the minimum length of a path of positive-capacity edges from s to v in G_f after i iterations
  - If there is no such path from s to v in G_f, then $d(v, i) = \infty$
  - Note that if $d(v, i) < \infty$ then $d(v, i) \leq |V| - 1$

The Main Lemma

- Lemma: For all v in V, and all $i \geq 1$, we have $d(v, i-1) \leq d(v, i)$
  - Suppose the claim fails for the first time after i iterations
  - Let f (resp., f') denote the flow at the start (resp., end) of iteration i
  - Let v be a vertex in $\{v \mid d(v, i-1) > d(v, i)\}$ minimizing $d(v, i)$, and let k denote $d(v, i)$
  - Observe that $k > 0$ since $v \neq s$
  - Let u be the predecessor of v on some shortest (and hence length k) path of positive-capacity edges from s to v in $G_{f'}$
  - Thus $d(u, i) = k - 1$ and hence $d(u, i-1) \leq k - 1$ by the definition of v

The Main Lemma (cont'd)

- The definition of v implies that $d(v, i-1) > d(v, i) = k$
- In the two cases below, we derive a contradiction by proving that $d(v, i-1) \leq k$
- Case 1: $c_f(u, v) > 0$
  - In this case, $d(v, i-1) \leq d(u, i-1) + 1 \leq k$
- Case 2: $c_f(u, v) = 0$
  - Since $c_{f'}(u, v) > 0$, edge (v, u) appears on the augmenting path of iteration i
  - Thus $d(v, i-1) = d(u, i-1) - 1$
  - Since $d(u, i-1) \leq k-1$, we have $d(v, i-1) \leq k-2$

Bounding the Number of Iterations

- At each iteration, the algorithm augments the flow f along some augmenting P in G_f
  - We refer to the edges of P with minimum capacity as "bottleneck" edges
- For any edge (u, v) in E, we will prove that the number of iterations in which (u, v) is a bottleneck edge is $O(|V|)$
- This implies that the total number of iterations is $O(|E| \cdot |V|)$
- Since each iteration can be performed in $O(|E|)$ time, the time complexity of the Edmonds-Karp algorithm is $O(|E|^2|V|)$

Bounding the Number of Times (u, v) is a Bottleneck Edge

- Suppose edge (u, v) is a bottleneck edge in iteration i
- Thus the residual capacity of edge (u, v) is zero at the end of iteration i
- Let k denote $d(u, i-1)$; thus $d(v, i-1) = k+1$
- The residual capacity of edge (u, v) remains zero until the end of the first iteration $i' > i$ for which the associated augmenting path P contains edge (v, u)
  - The edge (u, v) does not appear on the augmenting paths of iterations $i+1, \ldots, i'$
  - The main lemma implies that $d(v, i'-1) \geq d(v, i-1) = k+1$
  - Since (v, u) is on P, we have $d(u, i'-1) = d(v, i'-1) + 1$
  - Thus $d(u, i'-1) \geq k+2$

Bounding the Number of Times (u, v) is a Bottleneck Edge

- Suppose edge (u, v) is a bottleneck edge in iterations $i_1, \ldots, i_\ell$ where $\ell \geq 2$
  - We have shown that $d(u, i_j - 1) \geq d(u, i_{j-1} - 1) + 2$ for $2 \leq j \leq \ell$
  - Since $\ell \geq 2$ and $d(s, i) = 0$ for all i, we deduce that $u \neq s$
  - Since $u \neq s$, we have $d(u, i_1 - 1) \geq 1$
  - Thus $d(u, i_\ell - 1) \geq 2(\ell - 1) + 1 = 2\ell - 1$
  - Since (u, v) is a bottleneck edge in iteration i_ℓ, we know that $d(u, i_\ell - 1)$ is finite
  - The highest possible finite value of $d(u, i_\ell - 1)$ is $|V| - 1$
  - Thus $2\ell - 1 \leq |V| - 1$, implying that $\ell \leq |V|/2$
- Thus (u, v) is a bottleneck edge at most $\max(1, |V|/2)$ times

The Max-Flow Min-Cut Theorem

- Let G be a given flow network with real capacities
- Any execution of the Edmonds-Karp algorithm on G terminates after a finite number of steps with a maximum flow
- This establishes the existence of a maximum flow in G
- Given our earlier results, we obtain the following theorem, called the max-flow min-cut theorem
- Theorem: For any flow network G, a maximum flow exists in G and the value of a maximum flow in G is equal to the capacity of a minimum cut in G

Faster Maximum Flow Algorithms

- Later in the course we will study a faster maximum flow algorithm running in $O(|V|^3)$ time
- The best known bound is $O(|E| \cdot |V|)$
  - This bound is based on combining two algorithms, one for sparse graphs and one for dense graphs